8408-6086-RU000

LUNAR ORBIT RENDEZVOUS REFERENCE TRAJECTORY DATA PACKAGE (U)

JULY 15, 1965

(NASA-CR-154207) LUNAR ORBIT RENDEZVOUS REFERENCE TRAJECTORY DATA PACKAGE. VOLUME 1: APOLLO ERROR ANALYSIS. DESCRIPTION OF PROCEDURES AND COMPUTER PROGRAMS Final Report (TRW Systems, Redondo Beach, Calif.) 00/13 12404

N79-71951

Unclas

VOLUME I

APOLLO ERROR ANALYSIS — FINAL REPORT

Prepared under Contract No. 10001 to Bellcomm, Inc.

FACILITY FORM 602

(ACCESSION NUMBER)

27

(PAGES)

CR 71358

(NASA CR OR TMX OR AD NUMBER)

(THRU)

2A

(CODE)

30

(CATEGORY)

TRW SYSTEMS

AUG 1977

RECEIVED

NASA STI FACILITY

DISTRIBUTION

COMPLETE MEMORANDUM TO

CORRESPONDENCE FILES:

OFFICIAL FILE COPY
plus one white copy for each
additional case referenced

TECHNICAL LIBRARY (4)

NASA Hq.

Messrs. P. H. Bolger - MTE (2)
J. H. Disher - MLD
T. A. Keegan - MA (2)
D. R. Lord - MTD

MSC

Mr. C. C. Johnson - ET2 (2)

Bellcomm

Messrs. G. M. Anderson
C. Bidgood
W. T. Botner
B. F. Brown
J. O. Cappellari
D. E. Cassidy
J. P. Downs
S. S. Fineblum
D. R. Hagner
P. L. Havenstein
J. J. Hibbert
W. C. Hittinger
B. T. Howard
D. B. James
P. R. Knaff
H. S. London
J. Z. Menard
I. D. Nehama
G. T. Orrok
T. L. Powers
J. D. Richey
I. M. Ross
P. F. Sennewald
R. V. Sperry
W. Strack
T. H. Thompson
G. B. Troussoff
J. M. Tschirgi
C. M. Volk
R. L. Wagner

Department 1023 ← COPY TO

COVER SHEET ONLY TO

LUNAR ORBIT RENDEZVOUS REFERENCE TRAJECTORY DATA PACKAGE (U)

July 15, 1965

Volume 1

Apollo Error Analysis — Final Report

Description of Procedures and Computer Programs

Prepared by

F. J. Mullin

Under Contract No. 10D01 to Bellcomm, Inc.

under NASw 417

Approved by: B. H. Whalen

B. H. Whalen
Project Manager

Approved by: T. A. Magness

T. A. Magness, Manager
Space Guidance Department

TRW Systems
One Space Park
Redondo Beach, California

Available to NASA Offices and
NASA Centers Only.

CONTENTS

ILLUSTRATIONS

TABLES

1. INTRODUCTION

This report contains a description of the procedures and computer programs utilized in the generation of a linear Monte Carlo error analysis of the Apollo Mission. This error analysis was performed by TRW Systems Group for Bellcomm,Inc. under Subcontract 10001, Amendment 4. This volume, together with Volumes 1A, 1B, 1C, 4A, 4B, 4C, 5A, 5B, and 5C (STL documents 8408-6087-6101-RU000, respectively), Lunar Orbit Rendezvous Reference Trajectory Data Package, Apollo Error Analysis Final Report, constitutes the final deliverable item under Amendment 4.

A linear Monte Carlo error analysis is one in which a random sample of runs is made, each run consisting of the random generation and linear propagation of errors. The linearity is not in the generation of errors, but in the propagation of them. The error variances of some error sources are sometimes proportional to errors from other sources. Even if all error sources have Gaussian distributions, the resulting deviations are not Gaussian, due to these proportional relations. One should realize that the difficulty is not that errors are proportional to errors, but that error variances are proportional to errors. Since the deviations are not Gaussian, it is difficult to attach probabilities to one sigma, three sigma or k sigma tolerance regions. These probabilities can be estimated from the Monte Carlo sample.

The Monte Carlo method requires large sample sizes, however, and an integrating Monte Carlo simulation for the Apollo mission would be very costly in terms of computer time. This cost would be justified in preparation for an actual flight with a specified mission plan. However, as a mission planning and preliminary analysis tool, a linear analytic Monte Carlo simulation is more useful, since the required computing time is less by at least two orders of magnitude.

The Apollo Error Analysis Simulation is one in which an ensemble of Monte Carlo runs of the Command-Service Module (CSM) and the Lunar Excursion Module (LEM) are employed to compute statistical information which describes or summarizes the particular mission being simulated.

The general aspects of this Monte Carlo simulation have been described in Reference 1 which explains that the basic assumption of the error analysis is the existence of a reference trajectory from which all deviations are measured. If X_R denotes the reference state vector, X_A, the actual state vector and X_E, the estimated state vector, then by defining

$$\delta X_A = X_A - X_R$$
$$\delta X_E = X_E - X_R$$

it is possible to have the error analysis program work with the actual and estimated deviations δX_A and δX_E, rather than X_A and X_E. This considerably reduces the complexity of the equations which mathematically describe the operations of the CSM or LEM (since these deviations are assumed to propagate linearly) and makes it feasible for the Monte Carlo simulation to generate on the order of 1,000 samples which are to be used to compute the statistical outputs.

Reference 1 also explains that the LEM error analysis is run separately from the CSM error analysis. The reason for this is that the operations of the LEM are assumed to have no effect on the CSM since the CSM is passive in the rendezvous and docking maneuver. This allows the LEM simulation to be separate from that of the CSM. At the same time the CSM simulation does have a definite effect on the LEM simulation. The initial conditions for the LEM simulation are obtained from the CSM simulation since the CSM and LEM are attached prior to separation. Also the LEM simulation must know the terminal rendezvous conditions and obviously this requires a knowledge of the CSM deviations at the time of terminal rendezvous in order to compute LEM midcourse corrections as well as rendezvous. This means that the CSM simulation, sometimes referred to as the outer loop, must be run prior to the LEM simulation in order to have the necessary CSM information available. A more detailed description of exactly what is provided by the CSM simulation will be presented in Section 4 which describes the CSM simulation.

2. GENERAL DESCRIPTION OF THE ERROR ANALYSIS PROGRAM

The Monte Carlo simulation program, which is also known as TAPP VI, has as its output a magnetic tape containing the results of the individual simulations. These are to be used in producing the desired statistical quantities. The entire simulation is divided into a number of discrete events, each event being labeled or indexed in the order in which they occur in the simulation by an "i" value. There are essentially three kinds of events: 1) powered flight maneuvers, 2) impulsive midcourse corrections, and 3) free flight propagation with or without tracking information. At the start of each event, the simulation has an estimated deviation (measured from the reference trajectory), an actual deviation and a covariance matrix of the estimate. An event, starting at one fixed time and ending at another, uses the estimated deviation, the actual deviation, and the covariance matrix of the estimate at the start, to compute the same variables at the end of the event. At the same time, it takes into account what has occurred during the event. Reference 1 discusses three kinds of events in considerable detail. For the sake of completeness, the salient features of these are presented in Appendix A.

In the TAPP VI program the mathematical description of an event is called a PROP box and the various PROP boxes are indexed by a number P, P = 1, 2, . . . , n, where n is the number of different PROP boxes needed to describe the mission. For example, we construct a PROP box to generate the initial conditions for the simulation and set P = 1 for this box and let P = 2, 3, and 4 represent the PROP boxes used for powered flight maneuvers, impulse midcourse corrections, and free flight tracking and propagation, respectively. Then these four PROP boxes are sufficient to conduct the CSM Monte Carlo simulation from translunar injection up to but not including entry. To see this, let the sequence of events be indexed by i; then, the following table gives the sequence of events and the PROP box which describes the CSM simulation.

Table 1. CSM Sequence of Events and PROP Boxes

i	P	Comments
1	1	Generate initial conditions
2	2	Translunar (TL) Injection
3	4	Free Flight Propagation and Tracking (FFP and T)
4	3	First TL Midcourse Correction
5	4	FFP and T
6	3	Second TL Midcourse Correction
7	4	FFP and T
8	3	Third TL Midcourse Correction
9	4	FFP and T
10	2	Hyperbolic Deboost
11	4	FFP and T (Lunar operations)
12	2	Transearth (TE) Injection
13	4	FFP and T
14	3	First TE Midcourse Correction
15	4	FFP and T
16	3	Second TE Midcourse Correction
17	4	FFP and T
18	3	Third TE Midcourse Correction
19	4	FFP and T (up to entry)

The TAPP VI program then must call the PROP boxes in the proper sequence and cycle through the above sequence of PROP boxes a specified number of times in order to complete the Monte Carlo study. It must also write a tape of the individual samples which are the outputs of each PROP box. The logic necessary to accomplish this is shown in Figure 1 and is almost self explanatory. The individual Monte Carlo runs are indexed by k, with k_{max} being the specified number of runs. Table 1 (or its equivalent for the LEM simulation) is an input to the program, as is the largest value of $i = i_{max} = 19$ for the sequence given above in Table 1. The statistical processing part of the simulation will be discussed in a later part of this report.

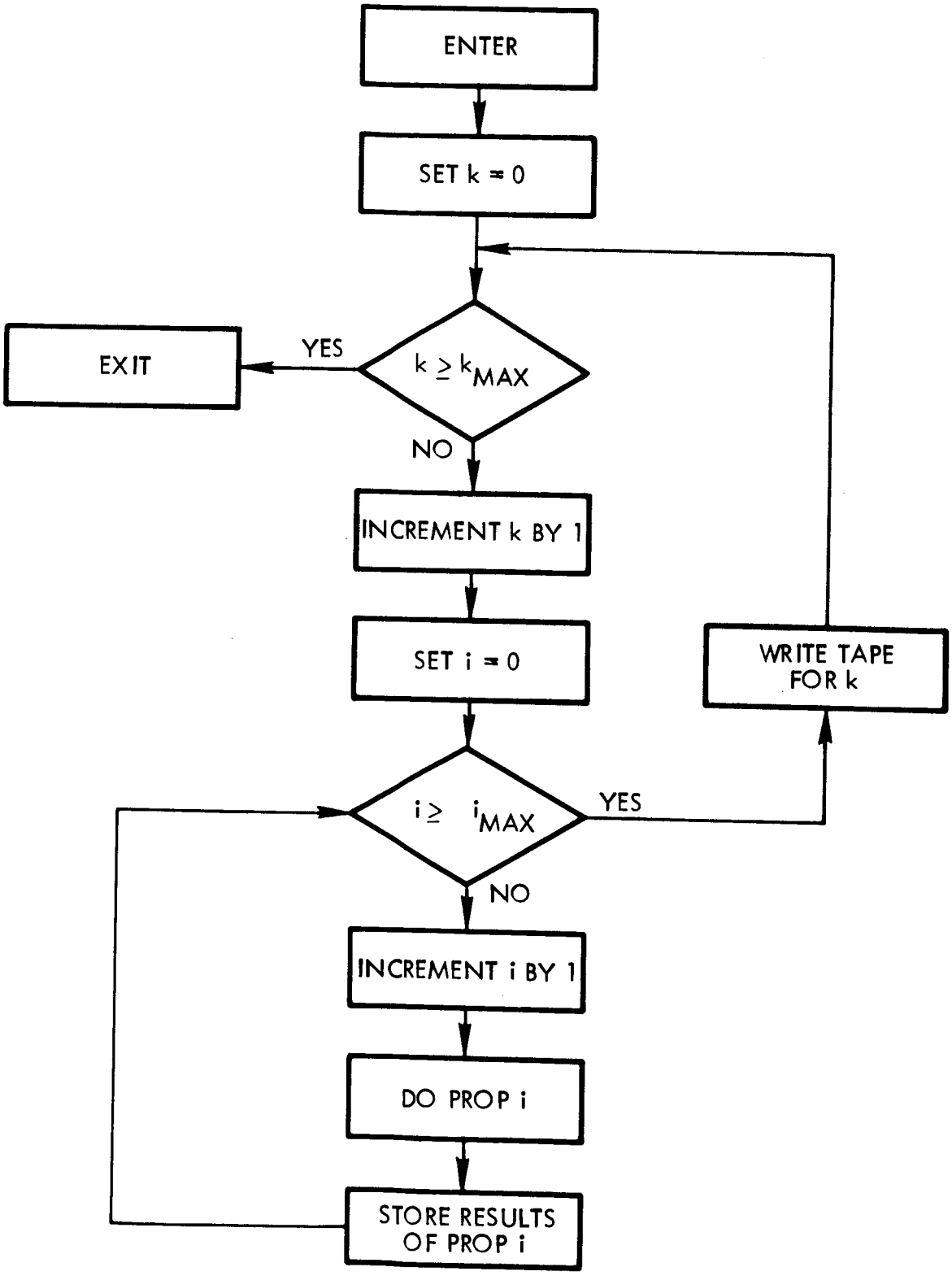

Figure 1. Logic Diagram of TAPP VI

It has already been mentioned that the CSM simulation and LEM simulation are run separately. The TAPP VI logic, however, is the same for both simulations, the only difference being the sequence of events and a few PROP boxes. The next section will briefly discuss the basic PROP boxes and following that a description of the two simulations will be presented. A description of the PROP boxes will assist the reader in understanding the description of the simulations.

3. BASIC PROP BOXES

In this section only the details of the PROP boxes will be discussed; the material of Appendix A and Reference 1 will justify the theoretical aspects of this presentation.

3.1 POWERED FLIGHT PROPAGATION

The deviations at the end of a powered flight phase ($\delta X_{A,1}$, $\delta X_{E,1}$) are assumed to be related to those at the start of the powered flight phase by

$$\delta X_{A,1} = \Phi_{AA}\,\delta X_{A,0} + \Phi_{AE}\,\delta X_{E,0} + \Phi_{AP}\,\delta P \tag{1}$$

$$\delta X_{E,1} = \Phi_{EE}\,\delta X_{E,0} + \Phi_{EP}\,\delta P. \tag{2}$$

The covariance matrix at the end of the phase is

$$\Sigma_{E,1} = \Phi_{AA}\,\Sigma_{E,0}\,\Phi_{AA}^{T} + \left(\Phi_{AP} - \Phi_{EP}\right)\Sigma_{\delta P}\left(\Phi_{AP} - \Phi_{EP}\right)^{T}. \tag{3}$$

The change in weight from the reference trajectory weight is given by

$$\Delta W = \Phi_{WA}\,\delta X_{A,0} + \Phi_{WE}\,\delta X_{E,0} + \Phi_{WP}\,\delta P. \tag{4}$$

δP is the performance parameter vector and has quantities like gyro drift rates, thrust variations from nominal values, etc., as its components. For this error analysis it is assumed that δP is a different random vector from what it was in the previous powered flight phase if the powered flight phases occur more than 20 hours apart. Assuming this to be the case, the PROP box for powered flight propagation is as shown in Figure 2. The quantity d is an input parameter and is equal to the dimension of δP. With this as an input quantity a different number of error sources can be considered in the various powered flight phases. The standard deviations of the individual components of δP are also inputs to the PROP box.

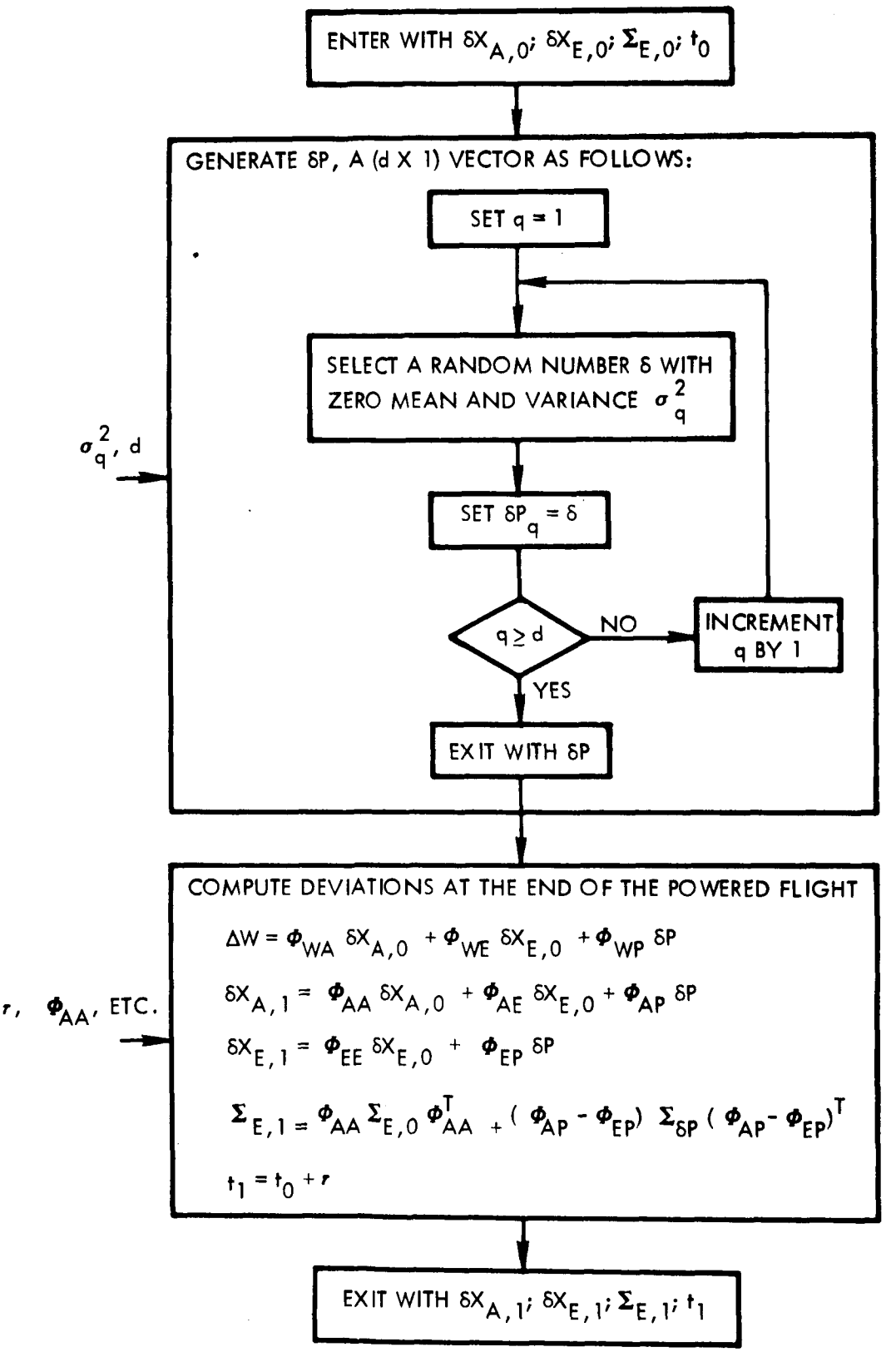

Figure 2. Diagram of Powered Flight Propagation PROP box

3.2 FREE FLIGHT PROPAGATION AND TRACKING

This PROP box takes an initial estimate and covariance matrix, propagates the estimate over the time interval under consideration (i.e., from the time at the start of the event to the time at the end of the event) and combines this updated estimate and its covariance matrix with the estimate formed from tracking information and the covariance matrix of the tracking estimate to arrive at a revised estimate and new covariance matrix. The propagated estimate and covariance matrix are given by

$$\delta X_P = \Phi \delta X_E \; ; \; \Sigma_P = \Phi \Sigma_E \Phi^T$$

The subscript P indicates this is a priori information. The tracking estimate in the simulation is given by

$$\delta X_T = \left(A^T W A \right)^{-1} A^T W \, r$$

where the A matrix is the partial of observations with respect to the variables being estimated; W is the weighting matrix and r, the residual, equals $A\delta X_A$ + n, where n is the measurement noise vector. The new estimate and covariance matrix are given by

$$\delta X_E = \left(\Sigma_P^{-1} + A^T W A \right)^{-1} \left(\Sigma_P^{-1} \, \delta X_P + A^T W A \delta X_A + A^T W \, n \right) \quad (5)$$

$$\Sigma_E = \left(\Sigma_P^{-1} + A^T W A \right)^{-1} \quad (6)$$

The only random part of Equation (5) is the term involving the measurement noise vector. In the simulation this part of the Equation (5) is generated by a random vector generating subroutine which produces a random $A^T W n$ vector which is assumed to have zero mean and a covariance matrix equal to $A^T W M W A$. $M = \overline{nn^T}$ and may or may not equal W^{-1}, according to the wishes of the person using the simulation program. If systematic errors, like station location errors, are to be included in the tracking model but not solved for in the estimation procedure, then Equation (5) should have the following term added to it.

$$\left(\Sigma_P^{-1} + A^T W A\right)^{-1} \left(A^T W B\, q\right)$$

The vector q is the systematic error vector and Bq is the term that is added to the residual, i.e., $r = A\delta X_A + Bq + n$ for this case. The PROP box for this event is shown in Figure 3.

3.3 IMPULSIVE MIDCOURSE CORRECTIONS

The midcourse correction error models for the CSM and LEM are explained in Reference 1. The essence of both error models is that a desired velocity correction is computed but the actual velocity correction applied to the spacecraft differs from the computed or desired velocity correction because of platform misalignment, accelerometer errors, and cutoff errors. These imperfections in the guidance system are selected randomly for each correction and are then used to compute the actual velocity correction. The mean squared value of these guidance system parameters is used to modify the estimate covariance matrix. The CSM and LEM error models are different because prior to each midcourse correction the CSM is assumed to realign its platform in such a way that one axis of the platform points in the direction of applied correction. The LEM does not necessarily realign its platform prior to corrections made along the ascent trajectory. Also the CSM uses its Reaction Control System to trim out the sensed error in the velocity correction which has been applied with the SM engine while the LEM uses only its Reaction Control System for corrections. A description of both velocity correction PROP boxes follows.

3.3.1 CSM Velocity Correction PROP Box

The PROP box used for CSM velocity corrections is shown in Figure 4. As indicated by this PROP box, the program first generates a random vector k whose first three components are the scale factor, bias, and nonlinearity of the accelerometer sensing acceleration in the direction of the applied velocity correction. The fourth and fifth components of k are the cutoff errors in feet per second of the high thrust (SPS) engine and the low thrust engine (RCS) and the last two components are angular errors in the yaw and pitch planes of the platform. The parameter b is a velocity bias which is used to help insure that the low thrust system essentially always adds to the correction applied by the high thrust

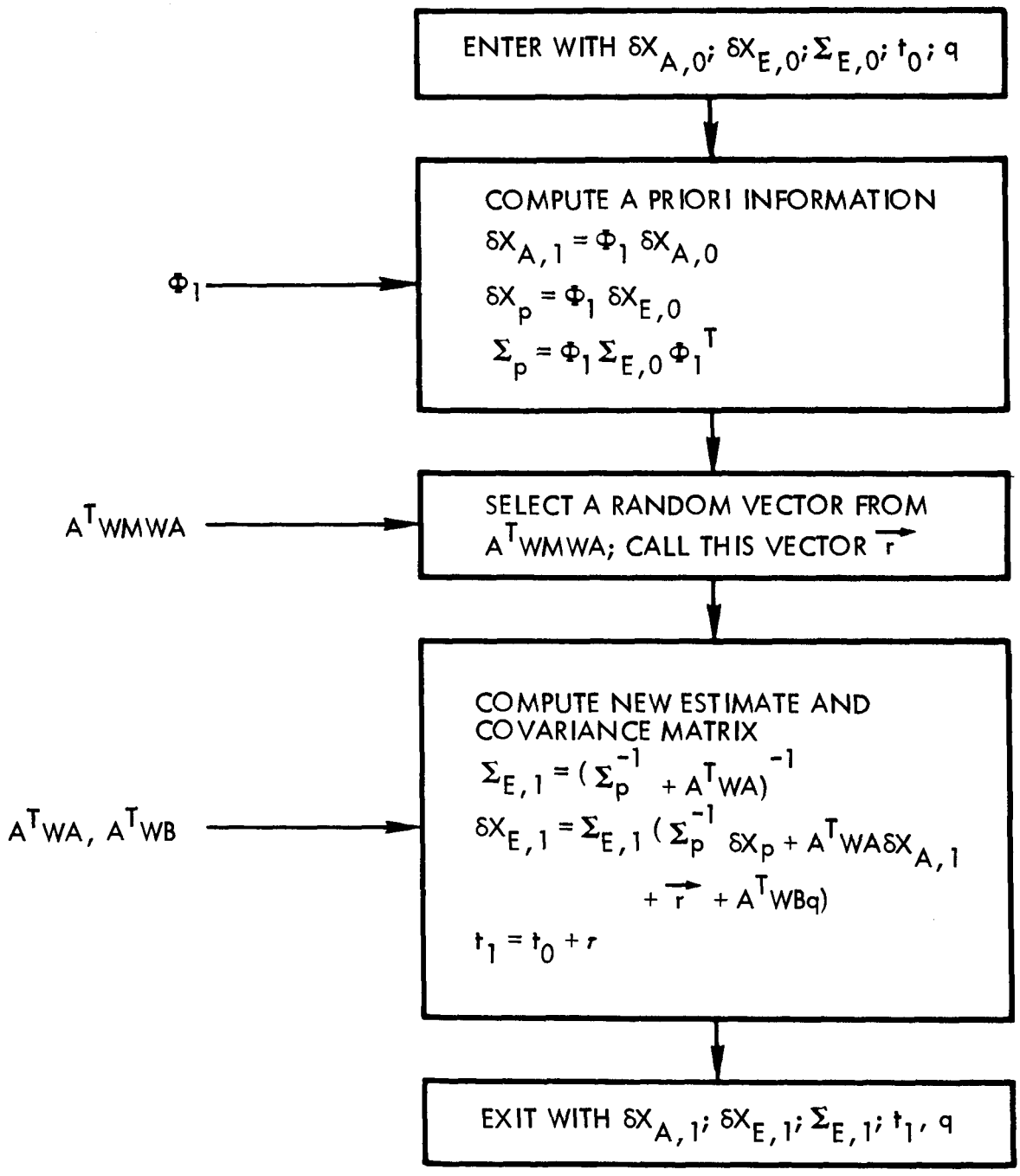

Figure 3. Diagram of Free Flight Propagation and Tracking PROP box

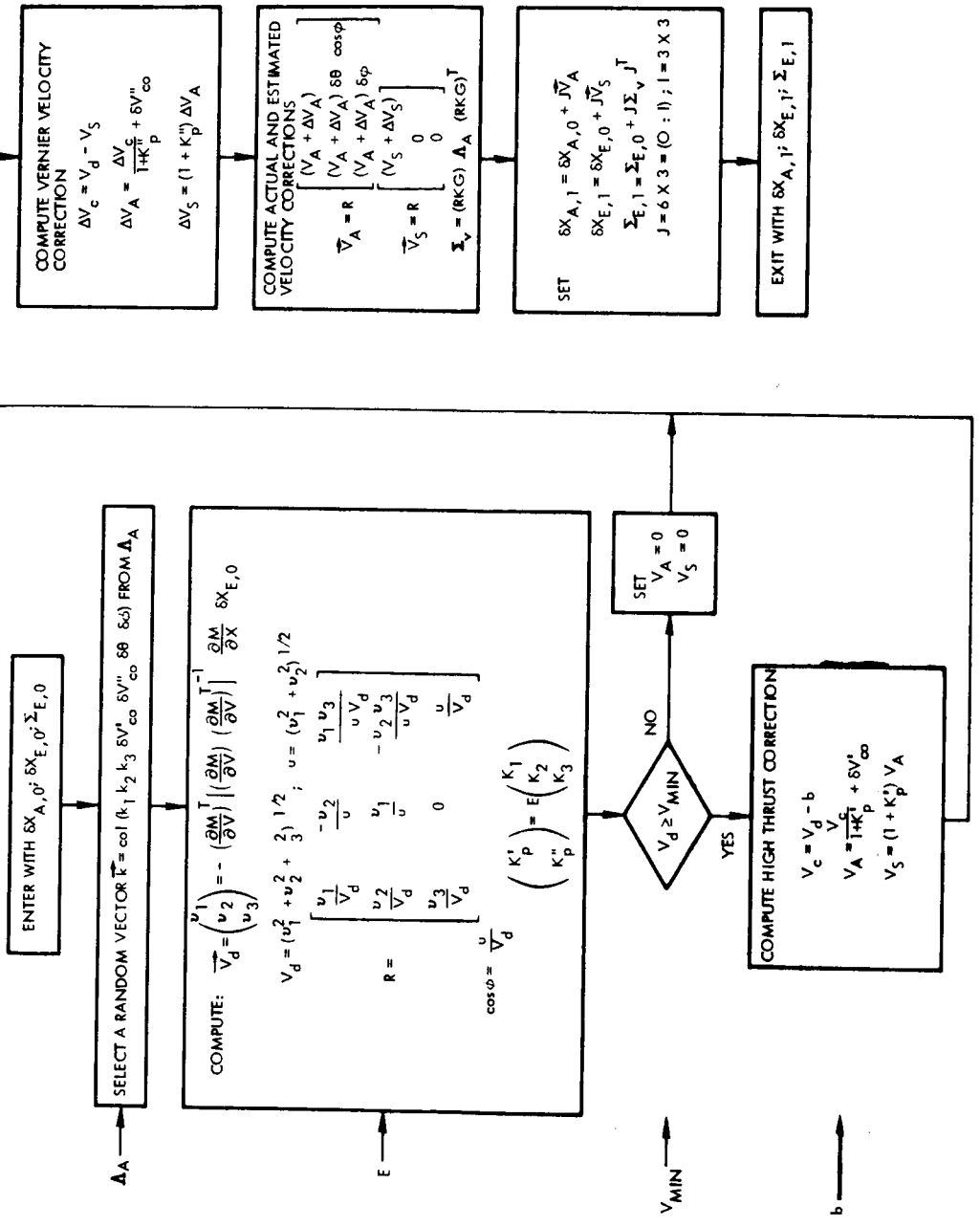

Figure 4. Diagram of CSM Midcourse Correction PROP box

system, thereby saving midcourse ΔV. The simulation computes the magnitude of the actual and sensed correction of the high thrust system, V_A and V_S, adds to this the actual and sensed correction of the low thrust system ΔV_A and ΔV_S and then determines the vector corrections utilizing the fact that one axis of the platform is aligned in the direction of the computed velocity correction. Finally the actual and estimated deviations and the covariance matrix are modified to complete the velocity correction phase which is assumed to take place in zero time.

3. 3. 2 LEM Velocity Correction PROP Box

The LEM Velocity PROP box is shown in Figure 5. The primary differences between this PROP box and that for the CSM velocity corrections are: 1) there is only one thrust level used to make corrections and 2) the rotation matrix, R, which relates the platform coordinate system to the inertial coordinate system, is expressed in terms of three small angular rotations rather than in terms of equivalent angular displacements in the yaw and pitch planes. This permits the elapsed time from the last platform alignment to be used directly as a parameter in the error model, and does not assume that the velocity correction is in the direction of one of the platform axes. This error model also assumes an impulsive correction takes place in zero time.

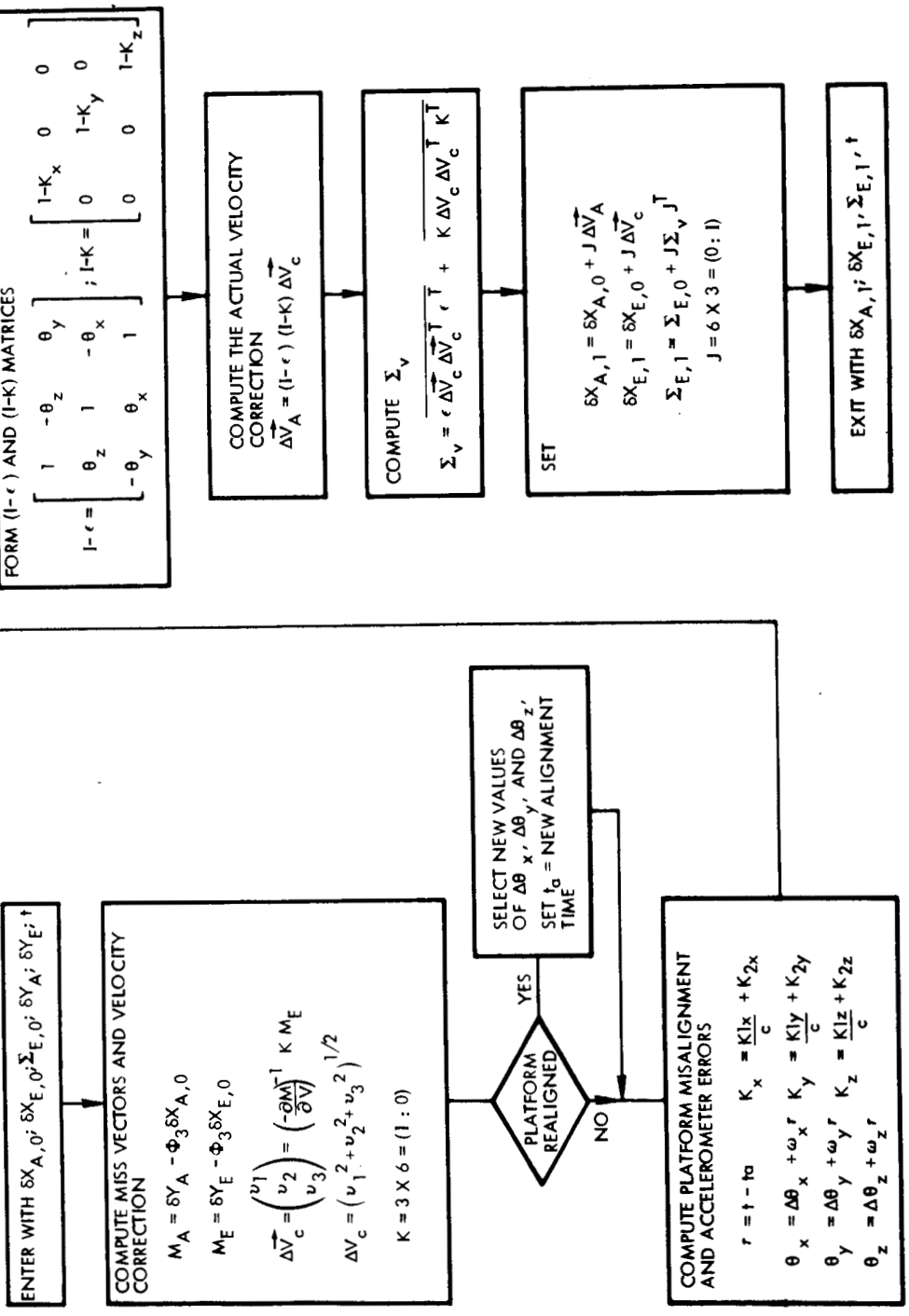

Figure 5. Diagram of LEM Midcourse Correction PROP box

4. THE CSM ERROR ANALYSIS SIMULATION

With this brief summary of the basic PROP boxes used in this error analysis it is possible to describe the details of the CSM simulation. This simulation actually starts at the end of translunar injection rather than prior to translunar injection as indicated in Table 1. The covariance matrix of actual deviations, used in randomly selecting the actual deviations at the end of translunar injection, was obtained by propagating the covariance matrix of the actual deviations at insertion into the earth parking orbit up to translunar injection using free flight partials. It is then propagated through translunar burn using powered flight sensitivity matrices. The estimate covariance matrix at the start of translunar injection was found by assuming that the MSFN network tracked the spacecraft while it was in the earth parking orbit, with no a priori information at the time of insertion into the parking orbit.

The actual deviation at the end of translunar injection was randomly selected from a multidimensional normal distribution whose covariance matrix was the covariance matrix of actual deviations. A second random vector, selected from a multidimensional normal distribution whose covariance matrix was the estimate covariance matrix, was added to the actual deviation vector to obtain the initial estimated deviation. The covariance matrix associated with the initial estimated deviation was found by propagating the estimate covariance matrix at the start of translunar injection through the translunar injection burn; this matrix was the same for all Monte Carlo runs.

Once the initial actual and estimated deviations are obtained they are propagated, along with the estimate covariance matrix, to the time at which the first midcourse correction is made and this propagated estimate and propagated covariance matrix as well as tracking data are used in determining the new estimate. A new estimate of the actual deviation and a new estimate covariance matrix are formed and are used in computing the velocity correction.

The objective of the first velocity correction is to force the spacecraft to be at a specified point on the moon's sphere of action at a specific time. This is a deterministic velocity correction (the miss vector is a three-dimensional vector) and customarily accounts for most of the ΔV expended on translunar

midcourse corrections. After this velocity correction, the estimated and actual deviations are propagated to the time at which the spacecraft arrives at the moon's sphere of action where a revised estimate is formed and a second velocity correction takes place. This process is repeated and a third velocity correction is made about 1. 5 hours prior to pericynthion passage. After the third midcourse corrections the deviations and covariance matrix are propagated to the time at which the hyperbolic deboost maneuver starts and, as a result of tracking data, accumulated during this time, a new estimate is formed.

Both the second and third midcourse corrections control only the radial and out of plane or normal deviations so that the miss vector is a two-dimensional quantity. The applied correction is the one which forces these deviations to zero and minimizes the magnitude of the velocity correction; this type of correction is sometimes called a "critical plane correction" (Reference 3). For those reference trajectories in which there is a plane change made during the hyperbolic deboost, the out of plane deviations are always measured normal to the desired lunar parking orbit plane, and not normal to the plane of the approach hyperbola.

The hyperbolic deboost phase follows and propagates the actual and estimated deviations as well as the estimate covariance matrix through this burn in the manner described in Section 3. 1. After this phase the deviations and covariance matrix are propagated (and revised using tracking data) to the time of Hohmann injection, and the conditions which exist at this time are recorded to be used as initial conditions in the LEM simulation. These same deviations are then propagated and updated with tracking information to the time at which the terminal rendezvous maneuver begins so that this information can also be used in the LEM simulation. Finally, the deviations and the estimate covariance matrix, at the time at which the terminal rendezvous maneuvers begin, are propagated and updated with tracking information to the start of transearth injection. The tracking data used in this simulation are combinations of on-board optical measurements of landmarks by the CSM and MSFN measurements of the range, range rate, azimuth, and elevation of the CSM. The manner by which various combinations of these measurements is available to the error analysis simulation will be described in Section 7 of this report.

Transearth injection is handled in the simulation in the same fashion as the hyperbolic deboost phase and the transearth coast phase is identical in structure to the translunar coast phase. There are three transearth velocity corrections, all of which try to control vacuum perigee altitude and out of plane deviations; consequently, all are critical plane corrections. After the third correction, the deviations are propagated to the time of entry on the reference trajectory at which time the simulation is completed. This entire procedure is repeated a specified number of times using the logic shown in Figure 1. After the completion of the desired number of Monte Carlo runs, the individual samples are processed to obtain the desired statistical output quantities. These are described in Section 6.

5. THE LEM ERROR ANALYSIS SIMULATION

With the description of the CSM simulation computed, it is possible to proceed with that of the LEM. The LEM simulation starts with the Hohmann injection phase and obtains its initial conditions from the CSM simulation. This means that the separation maneuver is assumed to introduce negligible errors and as such appears to be a reasonable supposition. With this as the starting point a table of events for this simulation could be written as follows:

Table 2. LEM Sequence of Events and PROP Boxes

i	P	Comments
1	1	Obtain initial conditions
2	2	Hohmann Injection
3	4	Hohmann Coast Phase (no tracking data)
4	2	LEM Powered Descent
5	5	Lunar Stay Period
6	2	LEM Ascent
7	4	Coasting Lunar Parking Orbit
8	3	Perform Plane Change and Initiate Rendezvous Trajectory
9	4	FFP and T
10	3	First LEM Midcourse Correction
11	4	FFP and T
12	3	Second LEM Midcourse Correction
13	4	FFP and T
14	3	Third LEM Midcourse Correction
15	4	FFP and T
16	6	Terminal Rendezvous Phase

Note that this simulation requires two additional PROP boxes, one for the Lunar Stay Period and the second for the Terminal Rendezvous Phase, each of which will be discussed in this section.

Once the initial conditions have been obtained, the powered flight PROP box is used for Hohmann injection. This is the same PROP box that was used in the CSM simulation for powered flight phases. Following Hohmann injection there was a free flight coast phase in which the deviations and covariance matrix are simply propagated to the start of LEM descent since there is no tracking data used by the LEM in this phase. At the end of this coast phase the LEM descent phase begins.

The powered flight sensitivity matrices for LEM descent and LEM ascent, which were used in this simulation, were provided by Bellcomm, Inc. The ascent sensitivity matrices are identical in form to those which have been previously described. This means that the standard powered flight PROP box is suitable for the phase. These sensitivities relate deviations at liftoff to deviations at a fixed time beyond the reference trajectory burnout time.

The LEM descent sensitivity matrices, however, are a little different. They relate deviations that exist at the time the LEM descent engine is started to deviations at the hover point. And since the criterion for starting the descent engine is based upon the LEM entering an imaginery cone centered on the landing site, it is necessary to provide in the simulation a variable time phase which terminates when the criterion for igniting the LEM descent engine is satisfied. By coupling this with the standard powered flight PROP box a pseudo fixed time to fixed time situation results.

There is, however, one other facit of this phase which distinguishes it from other powered flight phases. Bellcomm has chosen to provide sensitivity matrices which permit the calculation of the estimated deviation and the uncertainty in the estimate, $\delta X_u = \delta X_E - \delta X_A$, at hover rather then the actual and estimated deviations at hover. Also, the $(\Phi_{EP} - \Phi_{AP})\delta P$ term in the equation for δX_u is determined by selecting a random vector from a multidimensional normal distribution with a covariance matrix given by $(\Phi_{EP} - \Phi_{AP}) \sum_{\delta P} (\Phi_{EP} - \Phi_{AP})^T$ rather then by computing $(\Phi_{EP} - \Phi_{AP})\delta P$ directly. This covariance matrix was provided by Bellcomm in lieu of providing Φ_{EP} and Φ_{AP}. Thus, the PROP box for the LEM descent phase of the simulation is somewhat different from the PROP box used in other powered flight phases of the simulation.

This PROP box has a variable time phase at the beginning to account for the ellapsed time between the end of the Hohmann coast phase and the time when the criterion for descent engine ignition is satisfied. Once this criterion is satisfied, the simulation computes the estimated and actual deviations at hover using initial deviations and sensitivity matrices and a random vector to account for the effect of performance parameters.

Lastly, the simulation assumes that the LEM descends vertically to the surface of the moon from the hover point. This means the position components of the deviation state vectors at touchdown are the same as they were at hover but the velocity components of the deviation state vectors at touchdown are equal to zero. This is done because the simulation does not include any dynamics of the landing maneuver; the touchdown conditions are simply those at hover, with all velocity components set equal to zero.

Once the LEM is on the surface of the moon, it does nothing until liftoff time approaches and then the LEM tracks the CSM to determine its ephemeris. The LEM uses range, range rate, azimuth, and elevation measurements or combinations of these measurements to compute the ephemeris of the CSM; in addition, the simulation has the capability of accepting an estimate of the CSM ephemeris as determined in the CSM simulation and combining this with that computed in the LEM simulation. This covers the case in which the CSM would transmit this information to the LEM via a communication link and the LEM would in turn process it to obtain a revised estimate of the CSM ephemeris. All of the tracking measurements made by the LEM are relative measurements between the LEM and CSM and both uncertainties in the LEM touchdown position and biases in the measurements are considered in computing the CSM ephemeris.

The LEM tracks the CSM from the surface of the moon for one complete pass prior to liftoff and for that portion of the subsequent pass which occurs prior to actual liftoff. When liftoff does occur, the LEM has an estimate of the CSM ephemeris and with all future tracking information for the LEM, will try to determine its position relative to this ephemeris. In other words, while the LEM is on the surface of the moon it assumes that it knows its own position perfectly and determines the CSM ephemeris under this assumption. When it is in orbit, the LEM assumes it knows the CSM ephemeris perfectly and determines

its position and velocity accordingly. This means that it is necessary to determine a six-dimensional vector and not a twelve-dimensional vector in the onboard orbit determination procedure.

After the lunar stay period phase is completed, the LEM ascent phase starts. This phase uses the standard powered flight PROP box. At the end of this phase, the LEM is in a 50,000 foot parking orbit. The LEM coasts in this parking orbit taking range rate, azimuth, and elevation measurements between it and the CSM once every minute and continues doing this until the LEM enters the CSM orbit plane. This occurs roughly 90° away from the launch site since the LEM launch azimuth is parallel to the CSM orbit plane. Once the LEM enters the CSM orbit plane, an impulsive velocity correction is made to place the LEM on a rendezvous (i.e., collision) trajectory.

The manner in which this plane change and rendezvous maneuver is accomplished in the simulation is to insert a variable time phase between the end of the 50,000 foot parking orbit coast phase and the velocity correction phase. This variable time phase propagates the deviations until the estimated state vector is in the CSM orbit plane. A velocity correction is made, and a second variable time phase is then used to propagate the deviations and covariance matrix over a time interval. This interval depends upon the duration of the first variable time phase which makes the sum of the durations of the two variable time phases always equal to a constant time. This keeps the plane change maneuver in line with the fixed time to fixed time philosophy of the simulation.

This velocity correction, as well as the three later velocity correction, attempts to null the three-dimensional position miss vector at the nominal time of terminal rendezvous. The hardware errors associated with the inertial platform in the velocity correction phase are the same errors that were used in the ascent phase. It should also be pointed out that the LEM orbit determination scheme not only solves for the position and velocity deviations of the LEM but also for the biases in the radar measurements which constitute the tracking data.

After this last variable time phase there is a series of three tracking phases separated by velocity corrections. These velocity corrections are made

on a fixed time basis and the tracking measurements are still range rate, azimuth, and elevation. After the third midcourse correction, the deviations and covariance matrix are propagated and updated to some fixed time at which point the terminal rendezvous phase begins.

At the start of the terminal rendezvous simulation all deviations and covariance matrices (both LEM and CSM), which have been computed in an inertial coordinate system, are converted to relative deviations. The corresponding covariance matrix has its relative deviations expressed in orbit plane coordinates with the origin centered in the CSM. The axes of this orbit plane coordinate system are

$$\underline{1}_u = \frac{\vec{r}_{CSM}}{\vec{r}_{CSM}} \qquad \text{(radial direction)}$$

$$\underline{1}_v = \underline{1}_w \times \underline{1}_u \qquad \text{(tangential direction)}$$

$$\underline{1}_w = \frac{\vec{r}_{CSM} \times \vec{v}_{CSM}}{\vec{r}_{CSM} \times \vec{v}_{CSM}} \qquad \text{(normal direction)}.$$

The deviations are propagated and the same measurements are still made, until the estimated range between the LEM and CSM is 5 nautical miles. At this time a "velocity correction" is made to reduce the relative range rate between the two vehicles to roughly -60 feet per second. The velocity increment necessary to achieve this range rate is applied impulsively in the simulation. However, in order to more realistically simulate the non-zero burn time required to apply this velocity increment, the computer program computes the time that would be required to apply this velocity increment and propagates the deviations over this time interval, assuming that meanwhile no tracking information is gathered. There is also a short time delay added between the end of this phase and the start of tracking to account for any attitude orientations which may be necessary for radar visibility.

After the first terminal rendezvous velocity correction, the range rate measurements are replaced by range measurements; the angle measurement remain the same but measurement biases are no longer estimated. Measurements are again made every 60 seconds and this continues until the estimated range between vehicles is 1.5 nautical miles. At this time the range rate is reduced to -20 feet per second in a manner identical to that used for the first range rate reduction. After this, the deviations are propagated to a range of 0.25 nautical mile where the range rate is reduced to -5 feet per second. After this last velocity reduction the deviations are propagated until the relative range is 200 feet at which point the LEM simulation stops. If the relative range rate becomes positive at any time in the terminal rendezvous phase, the simulation proceeds to the next event which, generally speaking, will be a velocity correction phase and will result in an acceptable relative range rate which will allow the vehicles to rendezvous.

6. STATISTICAL PROCESSING OF SIMULATION DATA

The computer program, which is used to compute the desired statistical outputs, may be considered as a separate program for purposes of this discussion. The inputs to this program are the tape written by the error analysis simulation, an input table indicating what are the desired statistical quantities, and various matrices which may be necessary to make coordinate conversions. The simulation may do computations in one coordinate system and outputs may be desired in a different coordinate system.

In essence, the statistical processor will compute sample means and sample covariance matrices of the scalar and vector cumulative distribution functions. The sample mean is computed in the obvious manner, namely

$$\text{sample mean} = \frac{1}{N}\sum_{j=1}^{N} \underline{y}_j$$

$\underline{y}$ represents a specified vector produced by the simulation, the subscript j indexes the number of the Monte Carlo run, and N is the total number of runs. The sample covariance matrix is computed using the following Equation (4):

$$\text{Sample covariance matrix} = \frac{1}{N-1}\left[\sum \underline{y}_u \underline{y}_j^T - \frac{1}{N}\left(\sum y_j\right)\left(\sum y_j\right)^T\right] \qquad (4)$$

The normalized sample covariance matrix sometimes called the correlation matrix, is the above matrix modified by dividing the i,j^{th} element by $(\sigma_i\ \sigma_j)$ where σ_i is the square root of $(i,i)^{th}$ element of the sample covariance matrix.

The cumulative distribution routine first orders all of the samples in increasing numerical order and then produces a graph of the number of samples less than a specified number (the number of samples is normalized to unity) as a function of this specified number. This is done by plotting every 10^{th} sample

versus the numerical value of every 10^{th} sample. Actually, the cumulative distribution routine is more sophisticated than what has been described here. By simply noting that each sample increases the cumulative distribution function by (1/N), the routine has the capability of changing the increment of the distribution function between points so that one can, for example, study the tails of the distribution without having to print out an excessive number of points in the middle of the distribution.

The statistical processor can also produce sample means and covariance matrices of linear functions of the vectors generated in the simulation. This allows quantities to be computed in one coordinate system in the simulation and processed in a second coordinate system in the processor, but in general permits any desired linear function of the simulation quantities to be used in computing statistical information. Lastly, it should be noted that since the processor has a tape input, the processor can compute additional statistical quantities without re-running. This is done in the event that some additional statistical output is desired after the process or outputs are studied.

7. INPUT INFORMATION FOR THE ERROR ANALYSIS SIMULATION

It has already been mentioned that the Error Analysis Simulation Program is called TAPP VI. The letters TAPP stands for Tracking Accuracy and Prediction Program and the VI indicates that it is the sixth computer program developed in this series. Most of the input information for the TAPP VI program is contained on a tape written by a TAPP III program (5) which has received its input information from TAPP IV. (TAPP V is currently under development and, when available, will replace TAPP III.) In this section a description of the input information required for TAPP VI will be presented as well as a brief discussion of how this information is obtained from the TAPP III and TAPP IV programs.

The input information required for TAPP VI i.e., for the Error Analysis Simulation, is determined exclusively by the PROP boxes that are used in the particular simulation. TAPP VI must have all the inputs necessary to use the various PROP boxes at the proper time. This means 1) the powered flight sensitivity matrices for powered flight phases, 2) the state transition matrix, Φ, the tracking normal matrix, A^TWA, the A^TWMWA matrix (to be used in generating the random part of the estimate) and the A^TWB matrix (to account for unsolved systematic errors required for free flight propagation and tracking phases), and 3) the miss partials and related quantities for velocity correction phases. There is of course other input information necessary for TAPP VI which is not provided by TAPP III and TAPP IV. For example, the covariance matrices needed to generate initial conditions or the dimension of the performance parameter vector.

The TAPP III and IV programs provide all of what might be called trajectory type information for TAPP VI. This information being computed separately for each reference trajectory. The TAPP IV program computes the required trajectory information and TAPP III takes this information and puts it in the form necessary for use in TAPP VI. This is an over-simplification of what occurs but it does convey the general idea of the two programs.

To explain this kind of relationship in a little more detail let us consider just what is done in the TAPP IV program. For example the TAPP IV program

generates trajectory information at every one-half hour increment along the trajectory; that is, for every half hour along the reference trajectory TAPP IV produces an A^TWA, A^TWMWA, A^TWB, the state transition matrix, and the appropriate miss partials. This information is written on a tape and used as the input to TAPP III. Now, imagine that this information has been generated and that the first translunar midcourse correction occurs 5 hours after translunar injection. The TAPP III program will take all 11 of the trajectory points supplied by TAPP IV from $t = 0$ to $t = 5$ hours and produce one equivalent A^TWA, A^TWMWA, etc., which will enable the error analysis simulation to operate with the time from 0 to 5 hours being considered as the duration of a single event in TAPP VI. This also means that if it is desired to change the time of the midcourse correction from 5 hours to say, 10 hours, the trajectory type information necessary for this can be obtained by re-running TAPP III and having it "accumulate" the A^TWA matrices and the A^TWMWA matrices, etc., from $t = 0$ to $t = 10$ hours. It is not necessary to re-run TAPP IV.

The tape for the error analysis simulation which is provided by TAPP III also contains the powered flight sensitivity matrices. These matrices do not come from TAPP IV but rather are card inputs to TAPP III. TAPP III uses this information along with the TAPP IV to provide the simulation with all of the required trajectory information in the order in which it is needed. The trajectory information provided by TAPP III cannot be any finer in detail than that which corresponds to the times between TAPP IV outputs. This is one-half hour in the example under discussion.

The combination of TAPP III and TAPP IV allows the simulation to compute in whatever coordinate system (6) is most suitable for each phase of the mission; TAPP III provides the matrix to transform from the coordinate system being used to that desired for statistical outputs. This combination also allows the epoch of the tracking interval to be selected by the person conducting the error analysis simulation so that deviations can be expressed as those that exist at an arbitrary time rather than being restricted to deviations at the current time.

One other feature of these programs which should be referred to in this report concerns tracking data and the manner in which it is handled. TAPP IV

generates the A^TWA and other matrices for individual stations and/or individual types of measurements (e.g., range rate measurements) at a data range selected by the user. The classification or breakdown of the types of stations and kinds of measurements is also selected by the user of the program. This data is then given to TAPP III which will provide one A^TWA for the stations and data which are to be used in the TAPP VI simulation. For example, if one which to determine the effect of adding, say, range measurements on the accuracy of the tracking estimate, the tracking data would be separated into two groups in TAPP IV. One group would contain the A^TWA's, etc., for only range measurements and the second group would contain the same information for all measurements except range.

This would then be input to TAPP III which would produce (in turn) two sets of A^TWA's, etc., one with range measurements, and the second without range measurements. These would be used in two separate TAPP VI runs and by comparing the results in these two cases, the effect of the addition of the range measurements could be inferred.

8. SUMMARY

This report has described and discussed the Apollo Error Analysis Simulation Program. The basic routines (PROP boxes) used in computer program were presented and followed by a description of both the CSM and LEM error analyses and the program to produce statistical outputs. Finally, the interaction of the error analysis program with the current TAPP III and TAPP IV programs was outlined to complete the description of the Error Analysis Simulation.

REFERENCES

1. "Proposed Apollo Error Analysis Model and Associated Summary Format (U)," TRW/STL Report 8408-6082-RC000, December 14, 1964 (Confidential).

2. "Sensitivity Matrices for Apollo Error Analysis (U)," TRW/STL Report 8408-6084-RC000, February 15, 1965 (Confidential).

3. T. A. Magness, "Critical Plane Analysis," TRW/STL Interoffice Correspondence 9861.5-120, October 12, 1962.

4. A. Mood, "An Introduction to the Theory of Statistics," McGraw-Hill Book Company, 1950.

5. W. W. Lemmon, "Tracking Accuracy Matrix Processor (TAPP MOD III)," TRW/STL Report 8976-6027-RU000, December 11, 1963.

6. J. E. Brooks and W. W. Lemmon, "A Universal Formulation for Conic Trajectories: Basic Variables and Relationships," TRW/STL Report 3400-6019-TU000, February 1965.

APPENDIX A

A-1.0 Derivation of the Basic PROP Boxes

The purpose of this Appendix is to summarize the derivation of the basic PROP boxes which have been used in the error analysis simulation. This material has been discussed in Reference 1 and for additional information the reader is referred to that reference. In what follows these basic PROP boxes are discussed in functional terms, similar to what was done in Reference 1.

A-1.1 Powered Flight Propagation

In the error analysis it is assumed that the flow of information during a guided powered flight can be represented as shown in Figure 6. In this figure the integration routine contains a set of equations which are assumed to be an exact replica of the conditions the vehicle would experience in an actual flight. The output of this box is the actual state vector where state vector implies at least position, velocity and acceleration terms. This state vector is an input to a model of the sensing system, the sensing system being characterized by a sensing system parameter vector, P_s^*. The output of the sensing model represents the actual sensed quantities and is used as an input to the box which contains the navigation equations which in turn produce an estimate of the actual state. This estimate is used to generate guidance commands which affect the actual state vector.

A performance parameter vector P_p^*, is shown as an input to an integration routine and consists of terms like maximum thrust level, I_{sp}, etc. The vehicle performance parameter vector P_v may be formed by considering P_p and P_s as components of P_v; thus,

$$P_v = \begin{pmatrix} P_s \\ \\ P_p \end{pmatrix} .$$

In the following discussion the vehicle performance parameter vector will be used but it should be remembered that P_v consists of P_s and P_p.

*See Reference 2 for a complete list of the components of this vector.

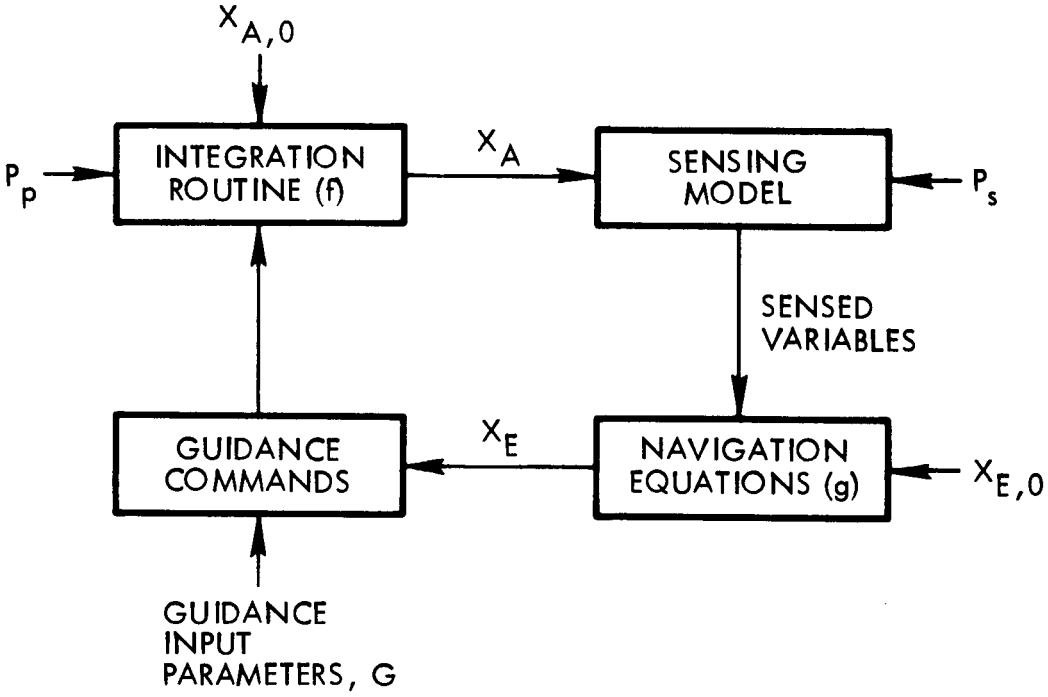

Figure 6. Information Flow Diagram for Powered Flight

If the subscripts 0 and 1 are used to denote quantities at the beginning and end of the powered flight, then in literal terms, the following equations can be written from Figure 6:

$$X_{A,1} = f(X_{A,0}, X_{E,0}, G, P_v) \tag{A1}$$

$$X_{E,1} = g(X_{E,0}, G, P_v) \tag{A2}$$

The variables used in Equation (A1) assume a guidance loop by involving both the estimated and actual initial states. Equation (A2) states that the navigation equations have a knowledge of only estimated states.

The reference trajectory may be defined to be the corresponding powered flight segment of the Reference Trajectory Data Package, Issue 4, as described in Reference 1, a so-called open loop reference trajectory; alternately, the reference trajectory may be defined to be a guided reference trajectory in which case the end conditions are arbitrarily close to those of the open loop reference when the actual on-board guidance equations are used. This guided reference trajectory is obtained by adjusting the guidance input parameters, G, until the end conditions are acceptably close to the end conditions of the open loop reference trajectory. The intermediate portion of the guided trajectory will differ from the intermediate portion of the open loop reference trajectory, since by definition, the open loop reference trajectory does not depend upon the state vector to compute guidance commands. Rather this open loop reference trajectory uses a predetermined function,which depends only upon time, to generate the equivalent of guidance commands. All powered flight sensitivity matrices for this error analysis were generated using guided reference trajectories.

The error analysis always computes deviations from reference for both the actual and estimated states. For powered flight segments, it was assumed that these deviations could be obtained by using the first order terms in a Taylor series expansion of Equations (A1) and (A2) about the reference trajectory. To first order accuracy the actual deviation can be written

$$\dagger\ \delta X_{A,1} = \left(\frac{\partial f}{\partial X_{A,0}}\right)\Bigg|_{\substack{X_{A,0} = X_{R,0} \\ X_{E,0} = X_{R,0}}} \delta X_{A,0} + \left(\frac{\partial f}{\partial X_{E,0}}\right)\Bigg|_{\substack{X_{A,0} = X_{R,0} \\ X_{E,0} = X_{R,0}}} \delta X_{E,0}$$

$$+ \left(\frac{\partial f}{\partial P}\right)\Bigg|_{\substack{P = \text{reference} \\ \text{trajectory} \\ \text{values} \\ G = G\ \text{reference}}} \delta P_v + \cdot \left(\frac{\partial f}{\partial G}\right)\Bigg|_{\substack{P = \text{reference} \\ \text{trajectory} \\ \text{values} \\ G = G\ \text{reference}}} \delta G \qquad \text{(A3)}$$

In this equation, δP_v represents the deviation of the vehicle performance vector from reference values , and δG represents the deviation of the guidance input parameters from nominal values.

Similarly,

$$\delta X_{E,1} = \left(\frac{\partial g}{\partial X_{E,0}}\right)\Bigg|_{X_{E,0} = X_{R,0}} \delta X_{E,0} + \left(\frac{\partial g}{\partial G}\right)\Bigg|_{G = G_{ref}} \partial G + \left(\frac{\partial p}{\partial P}\right)\Bigg|_{P = \text{ideal value}} \delta P \qquad \text{(A4)}$$

To avoid writing these two equations in the form given above, the following notation will be used.

---

† When f and x are vectors with elements $f_1 \ldots f_n$, and $x_1, \ldots x_m$, respectively, the symbol $\frac{\partial f}{\partial x}$ denotes an n x m matrix whose i, jth element is $\frac{\partial fi}{\partial xj}$. A vertical bar on the right side of the matrix indicates it is evaluated for specific values of its independent variables. These values are given below and to the right of the bar.

Let

$$\Phi_{AA} = \left(\frac{\partial f}{\partial X_{A,0}}\right)\Bigg|_{X_{A,0} = X_{R,0}} \quad ; \Phi_{A,E} = \left(\frac{\partial f}{\partial X_{E,0}}\right)\Bigg|_{X_{E,0} = X_{R,0}}$$

$$\Phi_{AP} = \left(\frac{\partial f}{\partial P}\right)\Bigg|_{P = \text{reference trajectory values}} \quad ; \Phi_{AG} = \left(\frac{\partial f}{\partial G}\right)\Bigg|_{G = G\,\text{reference}}$$

$$\Phi_{EE} = \left(\frac{\partial g}{\partial X_{E,0}}\right)\Bigg|_{X_{E,0} = X_{R,0}} \quad ; \Phi_{EG} = \left(\frac{\partial g}{\partial G}\right)\Bigg|_{G = G_{ref}} \quad ; \Phi_{EP} = \left(\frac{\partial g}{\partial P}\right)\Bigg|_{P = P_{ref}}$$

The first letter of the subscript on Φ indicates that the actual (A) or estimated (E) deviation is being considered and the second letter indicates the quantity that is affecting the deviation. With these definitions, Equations (A3) and (A4) can be rewritten in the following forms

$$\delta X_{A,1} = \Phi_{AA}\,\delta X_{A,0} + \Phi_{AE}\,\delta X_{E,0} + \Phi_{AP}\,\delta P + \Phi_{AG}\,\delta G \qquad \text{(A5)}$$

$$\delta X_{E,1} = \Phi_{EA}\,\delta X_{A,0} + \Phi_{EE}\,\delta X_{E,0} + \Phi_{EP}\,\delta P + \Phi_{EG}\,\delta G \qquad \text{(A6)}$$

In the analysis it is assumed that Φ_{EA} is the null matrix. It is shown in Equation (A6) only for completeness.

The various Φ matrices can be generated in a straight forward manner. For illustration imagine that Φ_{EE} is to be determined and the reference trajectory has been selected. All initial conditions are set equal to those used in generating the reference trajectory, except for the first component of $X_{E,O}$ which has some unit deviation added to it. With these intial conditions the simulation is now allowed to run. Cut-off then occurs and the actual and estimated states are propagated to a fixed time past the cut-off time in the reference trajectory. The difference between the new estimated state of this trajectory and that of the reference trajectory is computed and each component of this difference is divided by the unit deviation in $X_{E,O}$ to form the first column of Φ_{EE}. The same procedure is repeated with $X_{E,O} = X_{R,O}$ save for a unit deviation in the second component of $X_{E,O}$ and the second column of Φ_{EE} is generated. By continuing this procedure for all components of $X_{E,O}$ the complete Φ_{EE} matrix can be numerically determined. Note that the Φ_{EA} matrix can be computed at the same time with no additional computer runs since differences in the actual state can be used to generate Φ_{AE} column by column. Perturbations, component by component in P and G, then are made to generate the Φ_{EP}, Φ_{AP}, Φ_{EG} and Φ_{AG} matrices column by column. This fixed time past the cut-off time of the reference trajectory is chosen so that there is no danger of any trajectory having a burn time longer than this specified time. This allows a comparison of deviations to be made at a fixed time and results in the standard type of sensitivity matrices, i.e., sensitivities which relate deviations at one fixed time to deviations at a second fixed time. Figure 2 shows the PROP box used for powered flight propagation.

A-1.2 Free Flight Propagation and Tracking

At the start of any free flight section of this simulation the actual and estimated deviations are known, as well as the covariance matrix of the estimated deviation. Let the subscript (i-1) denote quantities at the start of the free flight segment and the subcript i refer to the same quantities at the end of the free flight segment. Thus, $\delta X_{A,i-1}$, $\delta X_{E,i-1}$ and $\Sigma_{E,i-1}$ are given and the problem is to compute $\delta X_{A,i}$, $\delta X_{E,i}$ and $\Sigma_{E,i}$, which is used as a priori information in forming a new estimate by the use of tracking data.

Since deviations from a reference are being propagated, it has been assumed that the standard technique of propagating deviations by linearizing the equations of motion about the reference trajectory is valid. Thus, it is assumed that

$$\delta X_{A,i} = \frac{\partial X_i}{\partial X_{i-1}} \delta X_{A,i-1} \tag{A7}$$

where the matrix $(\partial X_i / \partial X_{i-1})$ is analytically computed using two body partial derivatives, these partial derivatives being evaluated along the reference trajectory. Similarly,

$$\delta X_{E,i} = \frac{\partial X_i}{\partial X_{i-1}} \delta X_{E,i-1} \tag{A8}$$

where $(\partial X_i / \partial X_{i-1})$ is the same matrix as used in Equation (A7). Since the expected value of $\delta X_{E,i}$ is $\delta X_{A,i}$, that is

$$\Sigma_E (\delta X_{E,i}) \triangleq \overline{\delta X_{E,i}} = \delta X_{A,i},$$

the covariance matrix of $\delta X_{E,i}$, $\Sigma_{E,i}$, becomes

$$\Sigma_{E,i} = \overline{(\delta X_{E,i} - \delta X_{A,i})(\delta X_{E,i} - \delta X_{A,i})^T}$$

$$\Sigma_{E,i} = \left(\frac{\partial X_i}{\partial X_{i-1}}\right) \Sigma_{E,i-1} \left(\frac{\partial X_i}{\partial X_{i-1}}\right) \tag{A9}$$

Equations (A7) and (A8) and (A9) are the equations which were used for propagation in the simulation. The propagated estimated deviation and its covariance matrix is used as a priori information when a new estimate is to be formed.

To understand the formation of an estimate consider the following argument. Imagine that a trajectory is completely specified by a (n x 1) state vector X at some reference time. Suppose also that a noised set of observations z (k x 1) has been taken. Let f(X) represent the k rowed vector of observations that would have been made if the actual state vector were X and the observations were made with perfect accuracy. Then, the best (minimum variance) estimate of the state vector is that value of X for which the quadratic form

$$Q = \left[z - f(X)\right]^T W \left[z - f(X)\right] \tag{A10}$$

is a minimum.

W is the (k x k) matrix which weights the various observations and accounts for differences in accuracy of various measurements as well as for different types of measurements. One technique for determining the value of X for which Q is a minimum is to expand f(X) in a Taylor series about an initial guess X_o; thus, to first order

$$f(X) = f(X_o) + \left(\frac{\partial f}{\partial X}\right)\Bigg|_{X = X_o} (X - X_o) \tag{A11}$$

And by letting

$$r = z - f(X_o) \tag{A12}$$

$$A = \frac{\partial f}{\partial X}\Bigg|_{X = X_o} \tag{A13}$$

$$x = (X - X_o) \tag{A14}$$

Equation (A7) can be re-written

$$Q = (r - Ax)^T W (r - Ax) \tag{A15}$$

Note that a_{ij}, the element in the i^{th} row and j^{th} columns of the A matrix is the change in the i^{th} observation due to a variation in the j^{th} component of x. Now, the value of x for which Equation (A10) is a minimum will be denoted by $\hat{x}$ and can be written

$$\hat{x} = (A^T W A)^{-1} A^T W r \tag{A16}$$

This equation can be rewritten as

$$\hat{X} = X_o + (A^T W A)^{-1} A^T W r \tag{A17}$$

which means that the estimate of X_A equals the initial guess plus a term which comes from the least squares fit. The $\hat{x}$ given in Equation (A16) is a maximum likelihood estimate of X_A when the components of z obey a multidimensional normal distribution and W is the inverse covariance matrix of the measurement errors.

In the error analysis the matrix A is evaluated along the reference trajectory being used for the simulation, i.e., with $X_o = X_R$. All deviations are measured from the reference trajectory and the estimate of the deviation (δX_T) is determined by the tracking data. It can be written in terms of equation (A17) as follows

$$\delta X_T = X - X_o = X - X_R = (A^T W A)^{-1} A^T W r \tag{A18}$$

The covariance matrix of δX_T when W is the inverse of the noise covariance matrix is given by

$$\Sigma_T = \overline{(\delta X_T)(\delta X_T)^T} = (A^T W A)^{-1} \tag{A19}$$

Equation (A18) gives an estimate of the actual deviation of a trajectory from the reference and uses no a-priori information. In general, a second estimate and its covariance matrix is also available. This is found by

propagating the last estimate forward to the time corresponding to the state being estimated (see Equations A8 and A9). If this estimate and its covariance matrix are denoted by δX_P and Σ_P, the new estimate will be formed by combining δX_P and δX_T as though they were statistically independent. (The subscript P stands for a priori). This problem can be solved by finding the value of δX for which the quadratic form

$$Q = (\delta X_P - \delta X)^T \Sigma_P^{-1} (\delta X_P - \delta X) + (\delta X_T - \delta X)^T \Sigma_T^{-1} (\delta X_T - \delta X)$$

is a minimum. The value of δX for which this Q is a minimum is the best estimate of δX, denoted by δX_E, and is given by

$$\delta X_E = X_E - X_R = \Sigma_E \left[\Sigma_P^{-1} \delta X_P + (A^T W A) \delta X_T \right] \tag{A20}$$

where Σ_E, the covariance matrix of δX_E, is given by

$$\Sigma_E = \left[\Sigma_P^{-1} + A^T W A \right]^{-1} \tag{A21}$$

Note that it is not necessary to compute δX_T in order to determine δX_E; rather it is necessary to compute $(A^T W A) \delta X_T$ to determine δX_E. From Equation (A18)

$$(A^T W A) \delta X_T = A^T W r$$

Since the residual r can also be expressed as

$$r = z - f(X_R) = A \delta X_A + n \tag{A22}$$

Equation (A18) can be written for use in the simulation in the following form

$$(A^T W A)\, \delta X_T = A^T W\, r = A^T W A\, \delta X_A + A^T W\, n \qquad \text{(A23)}$$

Notice that the only random part of this expression is that involving the noise term $A^T Wn$. In the simulation, $(A^T WA)\, \delta X_T$ is computed by forming $(A^T WA)\, \delta X_A$ (δX_A is known for a particular simulation) and adding to it a random sample drawn from a population with a covariance matrix of $(A^T WMWA)$, where M is the covariance matrix of the noise. If $W = M^{-1}$, then this covariance matrix becomes $(A^T WA)$.

If there are some quantities or parameters which influence tracking data and which are not being estimated, the residuals, r, of Equation (A22) become

$$r = A\, \delta X_A + B\, q_A + n \qquad \text{(A24)}$$

where q_A represents the actual deviation vector of those quantities not being estimated and the matrix B represents the effect of these quantities on the observations. For this situation,

$$(A^T WA)\, \delta X_T = (A^T WA)\, \delta X_A + (A^T WB)\, q_A + A^T W\, n \qquad \text{(A25)}$$

The PROP box used for free flight propagation and tracking is shown in Figure 3.

A-1.3 CSM Velocity Corrections

For the phases involving CSM midcourse corrections the model for the error analysis is based upon an assumption of impulsive velocity additions. Figure 7 is a functional block diagram of the information flow in the simulation of such an impulsive maneuver.

The inputs to this block diagram from the previous phase are δX_A, δX_E and Σ_E. They are the deviations from reference of the actual and estimate

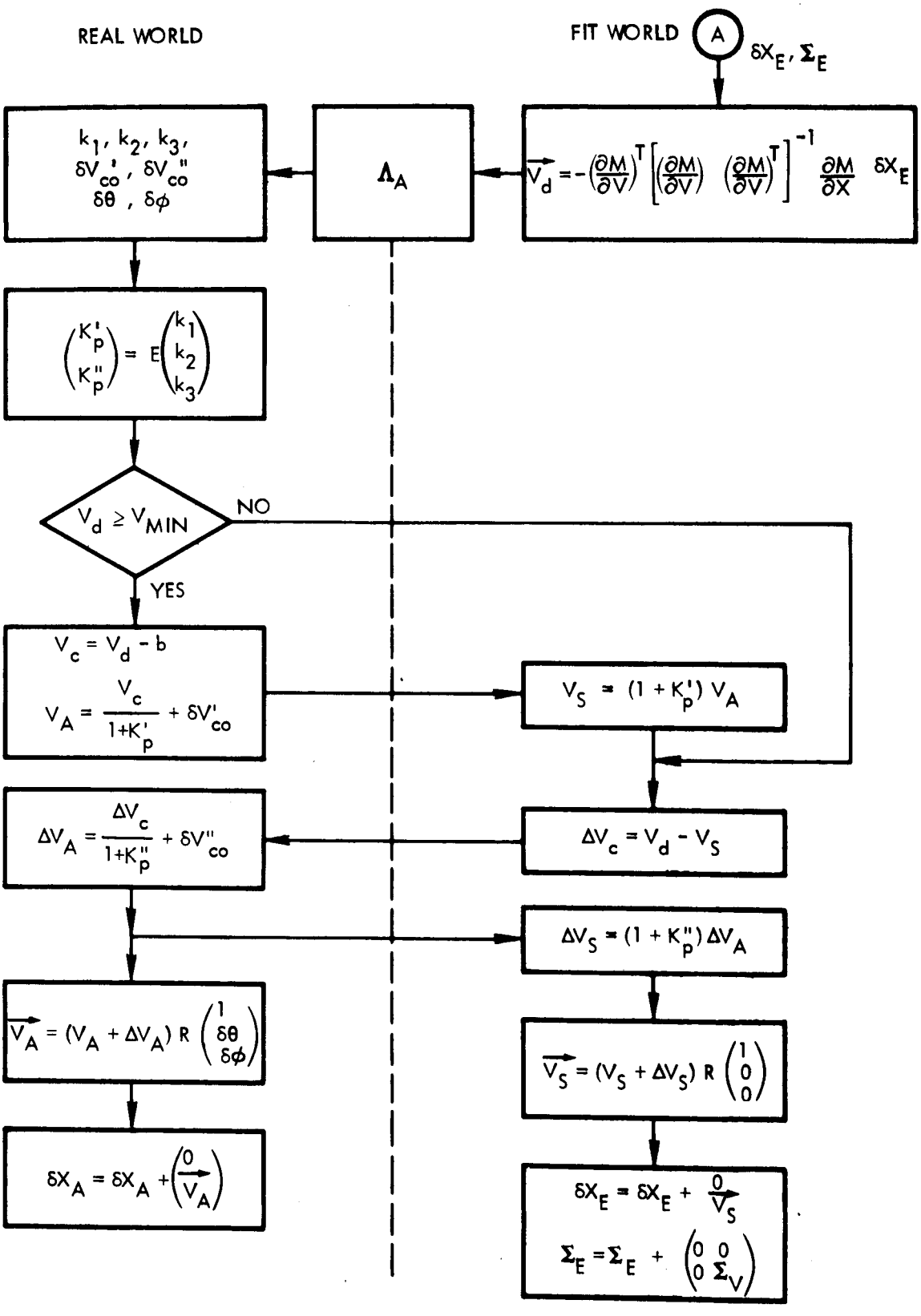

Figure 7. Information Flow Diagram for CSM Midcourse Corrections

state vectors and the covariance matrix of $(\delta X_E - \delta X_A)$, at the time prior to the maneuver. Each midcourse correction has a target vector associated with it, but its functional behavior is the same for all impulsive maneuvers. If the target vector is denoted as M, then the desired change in velocity $\overline{V}_d$ is computed as in block A. The matrix denotes the covariance matrix of the errors in the system execution (i.e., k_1, k_2, k_3, $\delta V'_{co}$, $\delta V''_{co}$, $\delta\theta$, $\delta\varphi$) where k_i, $i = 1, 2, 3$ denotes accelerometer bias, scale, and square terms, respectively. $\delta V'_{co}$ and $\delta V''_{co}$ denote the tail off velocity uncertanties associated with a high and low thrust systems. $\delta\theta$, $\delta\varphi$ represent errors in platform alignment at the time of the correction. For computational ease, K'_P and K''_P are computed from E where

$$E = \begin{bmatrix} \dfrac{1}{g_c\left(\dfrac{T}{w_o}\right)'} & 1 \left(\dfrac{T}{w_o}\right)' \\ \dfrac{1}{g_c\left(\dfrac{T}{w_o}\right)''} & 1 \left(\dfrac{T}{w_o}\right)'' \end{bmatrix}$$

$(T/w_o)'$ is the high thrust to initial weight ratio and $(T/w_o)'$ is the low thrust to initial weight ratio.

If V_d (the magnitude of $\overline{V}_d$) is larger than or equal to some minimum value, V_{min}, then the high thrust system is used and the low thrust system will act as a vernier correction. If V_d is smaller than V_{min} then the logic proceeds to the point where the low thrust system will attempt to achieve the desired velocity (V_d). The commanded velocity to the high thrust system is the desired velocity less some predetermined bias, b. The actual velocity acheived (V_A) during this high thrust phase is then computed, and the measured value of V_A, labeled V_s, is also computed. V_s is then used to determine the velocity required from the low thrust system. Note that if $V_s = 0$ the low thrust system will attempt to provide the desired velocity. In a manner

similar to that used with the high thrust system, the corresponding actual (ΔV_A) and measured (ΔV_s) velocities are then computed for the low thrust system. The components of the total actual velocity, $\overline{V}_A$, in the reference coordinate system are then obtained from a rotation matrix R and a velocity operator matrix

$$V_A + \Delta V_A = \begin{bmatrix} \begin{bmatrix} V_A + \Delta V_A \end{bmatrix} & 0 & 0 \\ 0 & (V_A + \Delta V_A)\cos\varphi & 0 \\ 0 & 0 & \begin{bmatrix} V_A + \Delta V_A \end{bmatrix} \end{bmatrix}$$

where φ is the attitude of the desired velocity with respect to the coordinate reference plane (xy plane). The components of the measured velocity $\overline{V}_s$ are generated in a similar fashion. Notice that in this case there are no angle errors ($\delta\varphi$ and $\delta\theta$) since the sensing system believes $\delta\varphi$ and $\delta\theta$ are zero. The final estimate and actual state vectors are then computed. Σ_V and the covariance matrix of ($\overline{V}_s - \overline{V}_A$) (which is added to the covariance matrix of the initial estimate to form the covariance matrix of ($\delta X_E - \delta X_A$) after the burn) is computed using

$$\Sigma_V = R\,K\,G\,\Lambda_A'\,G^T\,K^T\,R^T$$

These new matrices are defined below.

$$K = \begin{bmatrix} V_s & \Delta V_s & 0 & 0 \\ 0 & 0 & (V_s + \Delta V_s)\cos\varphi & 0 \\ 0 & 0 & 0 & V_s + \Delta V_s \end{bmatrix}$$

Λ_A' = the fit world estimate of Λ_A (Δ_A' includes both static and random effects).

$$G = \begin{bmatrix} E_{2x3} & 0_{2x2} & 0_{2x2} \\ 0_{2x3} & 0_{2x2} & 0_{2x2} \end{bmatrix}$$

A-1.4 LEM Velocity Corrections

The LEM midcourse correction error model is different from that of the CSM and can be developed in a relatively straightforward manner. The principal difference is that the LEM error model directly incorporates the effect of the time lapse between the platform alignment and the midcourse correction on the errors in the correction. In the LEM error model it is assumed that the midcourse velocity correction vector can have an arbitrary orientation with respect to the inertial platform axes. The matrix R which relates the platform axes to the (x, y, z) inertial reference frame is given by:

$$R = I - \epsilon; \quad \epsilon = \begin{bmatrix} 0 & \Delta\theta_z & -\Delta\theta_y \\ -\Delta\theta_z & 0 & \Delta\theta_x \\ \Delta\theta_y & -\Delta\theta_x & 0 \end{bmatrix}$$

The error model also assumes that there is a body mounted accelerometer outside of the control system loop which is used to cut off the RCS engine when the integrated acceleration equals the commanded velocity increment.

In the direction of the midcourse velocity increment, the error in the sensed acceleration of the accelerometer, used to cut off the RCS engine, is given by

$$\delta a'' = a''_{actual} - a''_{sinsed} = k''_1 + k''_2 \left(\frac{T}{W} g_c\right)$$

where k''_1 is the accelerometer bias error, k''_2 is the accelerometer scale factor error, and $(T/W)g_c$ is the acceleration. (The double primes indicate quantities in the direction of ΔV.) After some time interval, t, the error will be

$$\delta V_s'' = \Delta V''_{actual} - \Delta V''_{sensed} = \left[k_1'' + k_2''\left(\frac{T}{W} g_c\right)\right] t$$

However, the actual velocity is given by

$$\Delta V_a'' = \frac{T}{W} g_c t$$

Solving this equation for t and substituting it into the previous equation gives

$$\delta V_s'' = \left[k_1'' + k_2'' \frac{T}{W} g_c\right] \Delta V_a'' \frac{W}{TG_c} = \left[\frac{\frac{k_1''}{Tg_c}}{W} t\, k_2''\right] \Delta V_a''$$

Furthermore, it will be assumed that any uncommanded velocity increment associated with cutoff is small enough to be ignored and consequently, cutoff occurs when $\Delta V''_{sensed} = \Delta V_{computed}$. Hence

$$\Delta V_s'' = \Delta V_c \tag{A26}$$

$$\Delta V_a'' = \Delta V_s'' + \delta V_s'' = \Delta V_c + \left(\frac{\frac{k_1''}{Tg_c}}{W} + k_2''\right) \Delta V_a'';$$

$$\Delta V_a'' = \Delta V_c \left[1 - \frac{\frac{k_1''}{TG_c}}{W} + k_2''\right] = (1-k'')\, \Delta V_c \tag{A27}$$

$$k'' \triangleq \frac{\frac{k_1''}{Tg_c}}{W} + K_2''$$

The computed vector velocity correction is denoted by $\Delta\bar{V}_c$.

$$\Delta V_c = \begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix}; \quad \Delta V_c = \left[V_x^2 + V_y^2 + V_z^2\right]^{1/2}$$

The actual velocity increment can be expressed in the platform reference system (using primes to denote quantities in this coordinate system) as

$$\Delta\vec{V}'_a = (1+k'')\,\Delta\vec{V}_c \tag{A28}$$

By analogy with Equation (A27)

$$\Delta\vec{V}'_s = \left[1 - K'\right]\Delta\vec{V}'_a \tag{A29}$$

where

$$K'_{(3x3)} = \frac{1}{\frac{Tg_c}{W}}\begin{bmatrix} k'_{1x} & 0 & 0 \\ 0 & k'_{1y} & 0 \\ 0 & 0 & k'_{1z} \end{bmatrix} + \begin{bmatrix} k'_{2x} & 0 & 0 \\ 0 & k'_{26} & 0 \\ 0 & 0 & 1'_{2z} \end{bmatrix}$$

In this equation the k'_1 and k'_2 terms represent the bias errors and scale factor errors of the three platform accelerometers. Lastly, in terms of the true inertial reference frame the actual velocity, $\Delta\bar{V}_a$ is given by

$$\Delta\vec{V}_a = R\Delta\vec{V}'_a \tag{A30}$$

The error in the estimate of the applied velocity increment is given by

$$(\Delta\vec{V}_a - \Delta\vec{V}'_s) = \left[R - (I-K')\right] \Delta\vec{V}'_a = -(\epsilon - K')\,\Delta\vec{V}'_a \qquad \text{(A31)}$$

$$(\Delta\vec{V}_a - \Delta\vec{V}'_s) = -(\epsilon - K')(1+k'')\,\Delta\vec{V}_c$$

$$(\Delta\vec{V}_a - \Delta\vec{V}'_s) \approx -(\epsilon - K')\,\Delta\vec{V}_c$$

Finally,

$$\Delta V = \overline{(\vec{V}_a - \vec{V}'_s)(\vec{V}_a - \vec{V}'_s)^T} = \overline{\epsilon\,\Delta V_c\,\Delta V_c{}^T\,\epsilon^T}$$

$$+ \overline{K'\,\Delta V_c\,\Delta V_c{}^T\,K^T}$$

assuming there is no correlation between the platform alignment error and the accelerometer errors. Note that sensing errors associated with the accelerometer that cuts off the RCS engine are considered small and the effect of this error does not appear in the covariance matrix of ΔV. This error model also assumes that each accelerometer is correctly positioned with respect to its axis. Since this mounting error is expected to be small for any actual platform, its effect has been assumed to be negligible on sensed accelerations for this error model. The flow diagram of this error model is shown in Figure 8.

Checkout of the Error Analysis Programs

The TAPP series of programs described in this report individually generate a great deal of numerical information which must be verified if the results of the error analysis are to be meaningful. In checking out programs of this size it is impossible to look at every number produced or computed but is is possible to make some comprehansive tests on the programs to gain confidence in their ability to produce valid results.

The TAPP IV program supplies the basic trajectory data to TAPP III which in turn produces the input tape for TAPP VI. Thus, TAPP III essentially

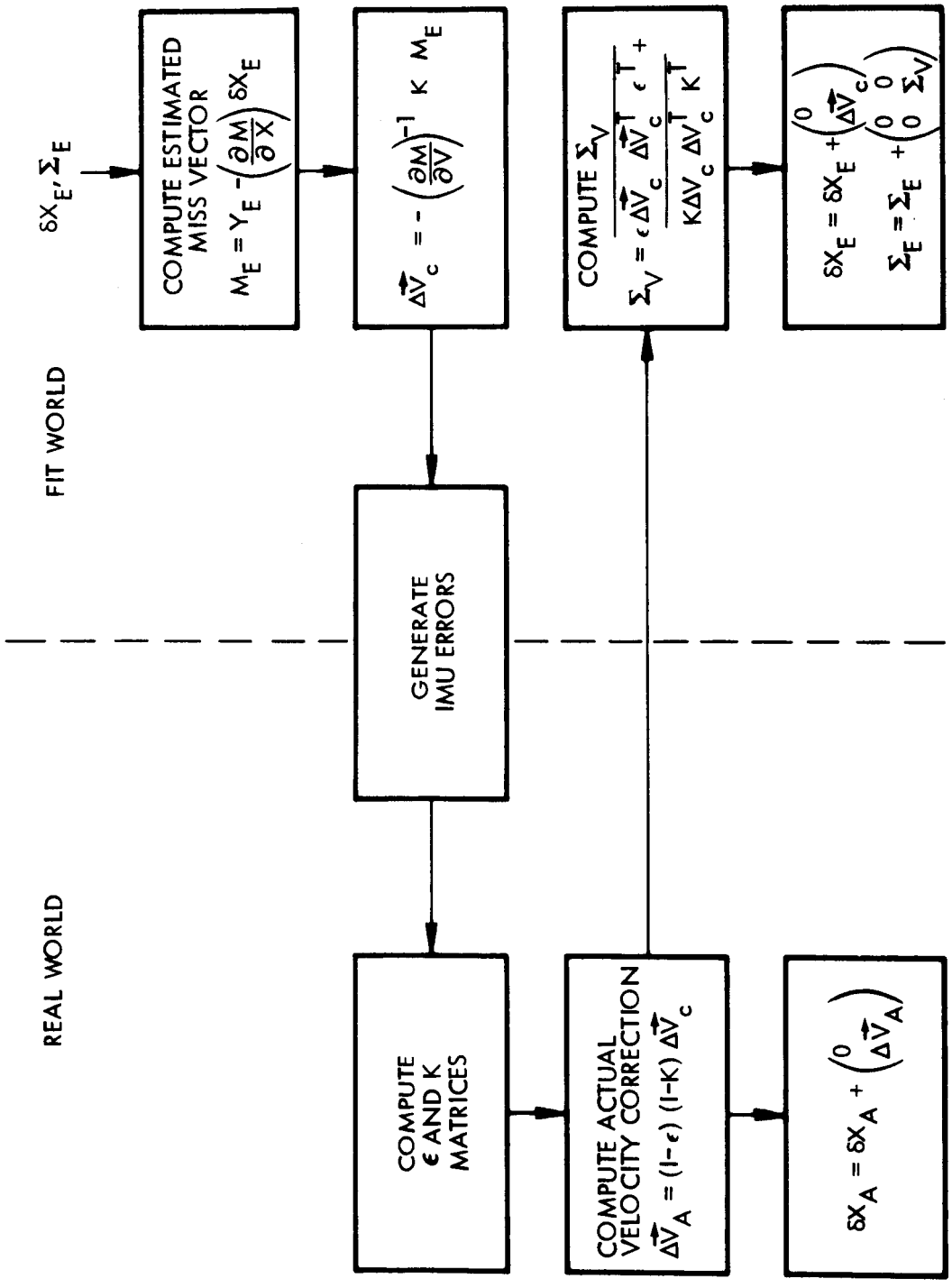

Figure 8. Information Flow Diagram for LEM Midcourse Corrections

reproduces data which is supplied to it and in this mode it is a relatively easy program to check. The input data to TAPP III is simply compared with the data on the tape produced by TAPP III, thus providing a convenient and complete test of the operational status of TAPP III. This is most easily done by not requiring TAPP III to make any coordinate changes in the data provided to it. Since the program goes through the same operations in all cases, the necessary transformation matrices (which are also provided by TAPP IV) are identity matrices. To further check TAPP III the powered flight sensitivity matrices, which are card inputs, can be checked to make certain these data cards are correct.

The checkout of TAPP IV was considerably more complicated than that of TAPP III, since this program provides the basic information for the error analysis. TAPP IV computes the reference state vector, the state transition matrices for both the LEM and CSM, the $A^T WA$ matrix and a coordinate conversion matrix which transforms the variables being used in the machine calculations to the coordinates in which the results are desired. The coordinate system used in the calculations was one of the α coordinate systems described in Reference 6 and the results were always specified to be in orbit plane coordinates.

The reference state vectors can be readily checked against those in the reference trajectory print-out. These checks will indicate that there is a small difference between the two corresponding vectors due to the fact that TAPP IV uses two body trajectories. The reference trajectory state vectors correspond to those computed using n-body equations of motion as well as the various harmonics of the earth and moon. Also, this comparison may require a coordinate conversion since the TAPP IV reference state vectors are expressed in geocentric or selenocentric coordinates while the reference trajectory state vectors are expressed geocentric and selenographic coordinates.

The state transition matrices computed by TAPP IV relate deviations at one time to deviations at a second time, and as such represent solutions to the variational equations. These matrices were spot checked by comparing the TAPP IV transition matrices with those produced by integrating programs which

have been operating for some time. Agreement in these matrices indicated that the TAPP IV computations were correct. It was not possible to check all of the transition matrices in this way. There are simply too many of them. Consequently, several of them were checked and when the TAPP IV results agreed with integrated results, it was assumed that all of the TAPP IV transition matrices were correct. Needless to say, great care was exercised to see that there were no input errors to the integration programs for the runs used to generate transition matrices.

The A^TWA matrices were checked in a manner similar to that used with state transition matrices. The A^TWA produced by TAPP IV for an earth parking orbit was compared with the A^TWA produced by the Generalized Tracking Program. This is an existing, operational program which can produce both normal matrices and covariance matrices. In addition, the standard deviations of the errors in the measurements, were all multiplied by the same number and the resulting A^TWA was checked with that obtained before multiplication to make certain that the A^TWA changed appropriately. Neither test guarantees that all of the A^TWA matrices computed by TAPP IV are correct but these tests and the results produced in the error analysis simulation, i.e., in TAPP VI, give a high confidence that all A^TWA's are correct.

The checkout of TAPP VI consists fundamentally of ascertaining that all of the PROP boxes are functioning properly and that all of the statistical information is being computed properly. This checkout was done at the same time TAPP IV was being de-bugged and was possible because of three things: (1) availability of a generalized matrix operation computer program, (2) the ability of being able to control the vectors produced by all random vector generating routines, and (3) a working and de-bugged terminal rendezvous simulation. The matrix operation program was used to duplicate the computations performed in the various PROP boxes with the same data as used by the PROP boxes. There was nothing random in this checkout since the numbers produced by the random vector generating routine were controlled and hence known. They were accounted for in both the PROP boxes and the matrix operation program.

The terminal rendezvous simulation was used to provide input data to the statistical processor and the results of two complete runs of this simulation were used to compute sample means and sample covariance matrices. These same sample means and sample covariance matrices were hand computed and in some cases, reproduced using the matrix operation program. When the results from the statistical processor agreed with the hand computed results, that particular statistical computation was assumed to be correct. The random vector generating routine was not controlled in the terminal rendezvous simulation runs so the inputs to the statistical processor were representative of what was to be expected from TAPP VI. In addition there were two other standard factors which were also used in checking these programs. These were the technical judgement and experience of the people involved in the ckeckout. Quite often it was some combination of these two intangibles which found some of the more elusive errors in the programs.

In summary, it can be said that while it was not possible to check everything in all of the programs, enough of each was checked to indicate that the individual programs were operating properly. This in turn produces a high confidence level in the numerical results of the error analysis.